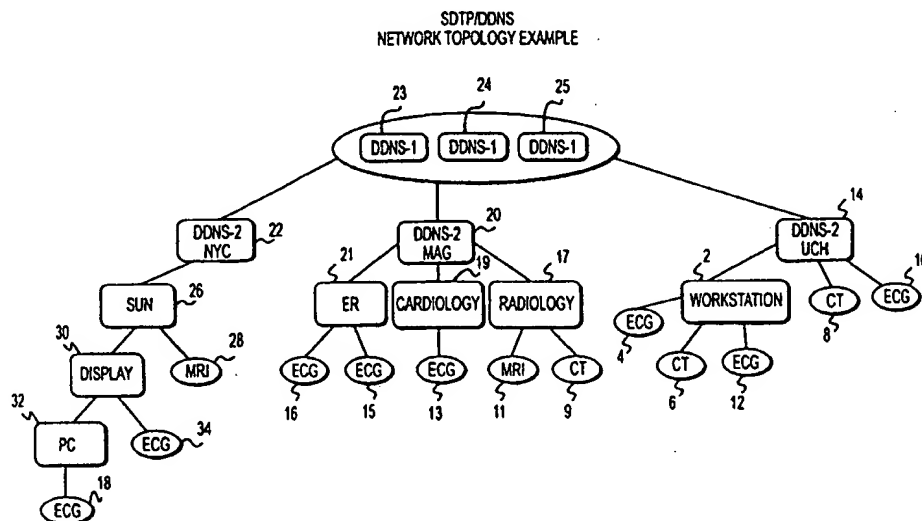


PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : H04L 29/12, G06F 19/00 // 159:00		A1	(11) International Publication Number: WO 00/03526
			(43) International Publication Date: 20 January 2000 (20.01.00)
(21) International Application Number: PCT/US99/14553		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 25 June 1999 (25.06.99)			
(30) Priority Data: 09/111,896 8 July 1998 (08.07.98) US			
(71)(72) Applicants and Inventors: OVERTON, John [US/US]; 5825 S. Blackstone, Chicago, IL 60637 (US). ROIZEN, Michael, F. [US/US]; 5622 South Woodlawn, Chicago, IL 60637 (US).			
(74) Agents: ROBERTS, Jon, L. et al.; Roberts Abokhair & Mardula, LLC, Suite 1000, 11800 Sunrise Valley Drive, Reston, VA 20191 (US).		Published With international search report.	

(54) Title: SYSTEM AND METHOD FOR ESTABLISHING AND RETRIEVING DATA BASED ON GLOBAL INDICES



(57) Abstract

A system and method for establishing and retrieving data based upon global indices established on the date of first use by a user. The system uses a distributed data name service (DDNS) which uniquely identifies users of the system based upon the unique ID of devices on the system combined with the date and time of first use by the user. Both the unique device ID and the unique user ID's are stored on servers of the system. This unique user ID generated is used for all subsequent uses of the system by the user. Searching for data generated by devices of the system relating to a particular user is accomplished by searching for instances of the user ID on servers of the system rather than by searching for the data itself. A simplified data transport protocol allows for the transfer of data from one location to another after the data is located.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

Title: System and Method for Establishing and Retrieving Data Based on Global Indices

Field of the invention

This invention relates generally to storage to data and retrieval of records. More particularly this invention relates to a universal method for generating an index of medical records at the time services are rendered and retrieving medical records based upon the automated indexing performed.

Background of the invention

Medical records can reside in many different places. As a patient sees different doctors and is treated for different conditions, individual records relating to the patient are created in each individual location. Therefore a medical record could exist at a general practitioner where the patient goes for annual physicals. At another time a medical record could be created for the patient at an immediate care facility where emergency room services are rendered. In a similar fashion a medical record for the same patient could be created at a particular specialist's office who treats the patient for a particular condition. All of these medical records may be critical to the treatment of the patient in any particular circumstance. If the total medical record for individual patient is not available, certain diagnoses may be overlooked or erroneously made.

Several systems have addressed the issue of how to create universal medical records. In general these systems create medical records by the creation of a file in some central storage area. Thereafter the central storage area may be accessed by individual practitioners by accessing the central storage of the medical record. Such systems use a "root registration" system wherein medical records and identities are registered centrally. Such systems generally are not fully automated leading to the potential for errors. Further only "registered"

1 records are available to remote users. Thus if a patient's medical record is not centrally
2 registered, it is simply not available to the practitioner.

3 Another disadvantage of the central registration process is that, at the present time, no
4 single format is universal. Thus many different medical organizations have different formats
5 which cannot be accessed among different medical institutions. Even if such access is
6 granted, format translation programs must be used which could cause additional errors in
7 translation.

8 One example of a system which attempts to obtain a master index of patient
9 identification information is the telemed system in use at Los Alamos labs. That system
10 maintains a master index of patient ID's thus tracking patient ID as a master reference. The
11 master ID is then used to determine where to find data related to a particular patient.
12 Telemed system deals with topological information normally characterizing patient records.
13 Further, the system relies upon "middleware" to resolve differences between database systems
14 that possess a particular patient's record. Thus a translation mechanism is necessary.
15 Further, the telemed system still requires a master patient index as a form of central
16 registration.

17 In contrast to the systems noted above, the present invention does not rely on a root
18 registration or a central registration of client information. Rather, the present invention
19 establishes an identity for a patient at the time of service, based on the identity of a given
20 device. This identity is established at the location of the device and not at any central
21 location. This identity is designated, however, in a universal fashion such that, for patient's
22 whose identity is established by the system, information relating to that particular patient can
23 be looked up in a convenient manner. Further, the present invention comprises the data
24 transfer protocol to allow for global addressing and retrieval of information from sites remote

from the location at which the patient is present. In this matter, all information concerning a particular patient maybe retrieved by the location treating the patient.

Brief description of the invention

It is therefore an objective of the present invention to be able to locate information by searching for indices of that information rather than for the information itself.

It is a further objective of the present invention to establish device driven unique identifiers that identify a person using the system.

It is a further objective of the present invention to establish device driven unique identifiers that identify a objects that are the subject of transactions using the system.

It is yet another objective of the present information to establish a global identifier for a user of the present invention the first time that a user uses the present invention.

It is yet another objective of the present information to establish a persistent identifier for a user of the present invention the first time that a user uses the present invention.

It is a further objective of the present invention to uniquely identify a particular device connected to the system.

It is yet another objective of the present invention to establish device identification the first time that a device is activated on the system.

It is a further objective of the present invention to make data universally available as soon as that data is created on the system.

It is another objective of the present invention to make data available at sites remote from the location at which the data is created as soon as the data is created.

It is yet another objective of the present invention to be able to search for data of interest without knowledge of the format in which the data was originally created.

It is a further objective of the present invention to allow local sites where data is

1 created to establish their own formatting and storage policies without such formatting and
2 storage policies being dictated by a central facility.

3 It is yet another objective of the present invention to establish security for a users
4 records by separating the user records from the identification card issues to a user.

5 The present invention uses a device-based paradigm to avoid the confusion and
6 restrictions associated with root registration systems. For example, a particular manufacturer
7 would register its company for participation with the present invention. Identification
8 numbers are assigned by the manufacturer and used to designate the equipment in question.
9 Thus equipment from a particular manufacturer which is used by a particular practitioner or
10 health care provider has a unique ID. A date/time stamp is also added to this equipment ID to
11 designate the source and when the equipment is used.

12 The present invention also comprises a simple network transport up protocol, defined
13 in this application as the Simple Data Transfer Protocol or SDTP. The SDTP provides
14 Internet wide sharing of data and database systems through a client/server. transaction based
15 model of data interaction and management. The SDTP allows for the transmission, reception,
16 and recovery of data from disparate locations.

17 The present invention also provides a network delivery mechanism for addressing
18 where to find requested information. This subsystem known as the distributed data name
19 service or DDN S is the reference system by which SDTP operates. This is not meant
20 however as a limitation. Once the location of information is established by the DDNS,
21 retrieval of information could occur equally as well by any protocol once that protocol knows
22 the location of the desired information.

23 Various universal encoding systems are also used in the present invention so that
24 individual devices and users of those devices can be encoded in a universal fashion.

1 It should be noted that the preferred embodiment illustrated in this specification is that
2 of a medical device and data retrieval system. However, the present invention should not be
3 construed as being so limited. For example the architectures and topology of the present
4 invention is equally well suited to commercial transactions such as point of sale transactions,
5 the generation of ATM cards and other commercial ventures. It can also be used as a form of
6 identification for employees of large organizations where security and access to facilities in
7 disparate locations must be tightly controlled. Thus while the medical application will be
8 elucidated below, those skilled in the art will appreciate that the system and method described
9 can be applied in many disparate situations.

10 Using the present invention, device manufacturers register their own unique names
11 with the system. For example Hewlett-Packard may register the name "HP" to be used with
12 all of its device ID numbers whereas another manufacturer such as Phillips might register
13 another different name "PH." When a medical device is first used on the system, its own
14 identification (manufacturer, and device ID number) is automatically registered with the
15 system. A patient receives an ID number, the first time the patient receives an ECG or is x-
16 rayed by equipment that is registered with the system of the present invention. Thus the first
17 time a patient is examined via medical equipment of the present invention, a universal record
18 of not only the equipment, but also the patient is automatically created.

19 The present invention also comprises a user identification token or barcode label
20 placed on any card of any variety which may be in the form of a credit card, smart card or
21 other token card which is generated at the time of the first use of any device registered with
22 the system. From that point on, the patient identification card is "registered" within the
23 system. Further, the health care provider simply uses the imaging equipment available based
24 upon the patient identification card, the universal encoding of the system, and images

1 recorded with appropriate patient identification information and image information. In
2 addition, the image created is universally available immediately after creation.

3 After recording information, the present invention tracks indices to locations of
4 information. If the location of the device changes, that information could be tracked by the
5 DDNS level 1 server so that queries could be automatically rerouted to the location at which
6 the device is currently housed. Thus the information can change as necessary while the index
7 to access that information does not. The present invention also does not require standardized
8 formatting of information. Thus, local sites format and store their own information as they
9 desire without having to adhere to a particular dictated format. Thus local sites do not have
10 equipment, staffing, administration, and other matters imposed upon them.

11 The system of the present invention transports, sends, delivers, receives, and
12 processes information objects. No middleware is required. Transmission of request for
13 information and the receiving of that information is done, using the simplified data transfer
14 protocol. In addition, existing systems can be included within the present invention since any
15 kind of document or object can fit within the present invention as an object. Only
16 namespaces as addresses are necessary for the present invention in order to find the location
17 of desired information and retrieve that information.

18 In summary, the present invention is a device driven addressing system rather than a
19 top-down addressing system. Individual devices create the namespace address necessary to
20 retrieve information created by the individual device. Thus the present invention allows the
21 minimal set of possible information at the top-level, which is used for routing requests for
22 information, with actual information created by individual devices or sites stored and located
23 at those devices or sites.

24 The present invention comprises a Simple Data Transport Protocol (SDTP),

1 Distributed Data Name Service (DDNS) software implementation, and a paradigm for
2 automated indexing of global databases.

3 The DDNS design is similar in function to the domain name service which supports
4 all Internet addresses. The domain name service for the Internet allows a single address to be
5 used by any user regardless of that user's location to find another user on the Internet. In a
6 similar fashion the DDNS of the present invention supports such a lookup service. However
7 DDNS is generalized and optimized for resolving database locations and database service
8 locations.

9 The DDNS exists in a series of servers in a tree structure whereby medical diagnostic
10 equipment are connected to servers. These lower level servers are in turn connected to higher
11 level servers in a tree structure or parent-child relationship. There is no practical limit to the
12 level of servers in the tree structure. It is only required that there be sufficient levels of
13 servers to satisfy the query needs of the organizations connected to the DDNS network.

14 Using the DDNS of the present invention, if a client machine requires information it
15 does not have, it sends a query to a parent server concerning where to find the record
16 information. In this application, the parent server is referred to as a DDNS Level 2 server or
17 DDNS-2 server. This situation can exist in the medical sense if a patient, having a medical
18 ID card of the present invention, visits an emergency room in other than the patient's home
19 city. In that situation the patient may use the patient ID card which will not be recognized by
20 the local medical diagnostic equipment. In that case the medical diagnostic equipment will
21 query the next higher server regarding where to find information on the patient.

22 If the server being queried has the necessary information, and answers the requesting
23 client, the Interaction stops. If the server does not have the information, it in turn, asks its
24 parent server, and so on up a tree structure of parent-child DDNS servers until the requested

1 information is found. Once the patient index information is found, it is passed back down to
2 the originating client which receives address/index information for a direct site to site request.
3 At this point a peer-to-peer connection can be made whereby the client receives the desired
4 medical information directly from the medical diagnostic equipment or database possessing
5 that information.

6 **The Simplified Data Transport Protocol (SDTP)**

7 Once the source of the desired information is located it becomes necessary to transfer
8 the desired information from one location to another. The Simplified Data Transport
9 Protocol (SDTP) of the present invention has this task. SDTP provides Internet-wide
10 sharing of data and database systems through a client-server, transaction-based model of data
11 interaction and management. SDTP structures transmission, reception, and recovery of data.

12 **Brief Description of the Figures**

13 Figure 1 illustrates the DDNS server nodes

14 Figure 2 illustrates a DDNS network topology

15 Figure 3 illustrates a specific example of an instance of use of the DDNS network topology

16 **Detailed Description of the Preferred Embodiment**

17 As noted earlier, the present invention comprises a Distributed Domain Name Service
18 (DDNS) software implementation whereby devices and users are uniquely identified and
19 registered on servers of the system the first time the devices are used and the first time that
20 users use the devices, a Simple Data Transport Protocol (SDTP) whereby once data of
21 interest is located, that data can be transported from location to location with ease, and a
22 paradigm for automated indexing of global databases.

23 **Distributed Data Name Service (DDNS)**

24 Distributed Data Name Service (DDNS) provides a name lookup service for the

1 indexing of global databases. It is designed to work in between a transfer protocol such as
2 SDTP, and an encoding scheme for naming objects uniquely.

3 The DDNS implementation is similar to DNS (Domain Name Service), which
4 supports all Internet name lookup service. The basic idea is illustrated in Figure 1 DDNS
5 Nodal Indexing.

6 In DDNS, if a client machine needs information it does not have, it asks a parent
7 server where to find that information. If that server has the information, it answers the
8 requesting client and the interaction stops. If the server does not have the information, it in
9 turn asks its parent server, and so on, up a tree structure of parent-child DDNS machines,
10 until the requested information is found. Once the information is found, it is passed back
11 down to the originating client, which receives forwarding information for a direct site-to-site
12 request, at which point a peer-to-peer connection is made "horizontally" in the tree structure.

13 It is important to note that, strictly speaking, DDNS is not "Middleware". Although it
14 can appropriately interact with Middleware as necessary.

15 DDNS provides efficient recovery of records from anywhere on a network, and has no
16 machine-type or operating system restrictions whatsoever. Its architecture provides intrinsic
17 scalability suitable for supporting universal databases that may require disk space exceeding
18 current technologies for individual sites. Since it resolves namespaces rather than IP
19 addresses, DDNS will seamlessly migrate to new network protocols such as "IP2" whenever
20 traditional IP is replaced. Using namespaces also supports organizational durability, since
21 organizations may change names and have these reticulated through the DDNS structure, or
22 keep the same names and change the undergirding machine hardware supporting those names
23 without impact of data accessibility to the network.

24 The SDTP/DDNS combination provides an automatic, low-level addressing and

1 retrieval mechanism on which other functionality can be conveniently built. Such
2 functionality may include automatic invoicing, automatically generated statistical polling of
3 part quality, demographic information for illnesses without access to patient identity, etc.
4 Such functionality supports electronic interaction and electronic commerce.

5 Used with SDTP and other naming and classification conventions, DDNS can provide
6 global indexing of any kind of image, produced on any kind of image-producing device,
7 making any image retrievable by a click of a barcode reader. Yet images may be produced on
8 different machines in different countries. Implications of such design specifically include
9 SDTP/DDNS/ASIA support for universal image recall through standard medical cards given
10 to patients at local hospitals. This is discussed specifically elsewhere in the document.

11 Used with other encoding schemes, SDTP/DDNS functionality may conveniently extend
12 into diverse applications. Such applications may include automated part tracking, automated
13 consumer purchase and repurchase, automated manufacturer-retailer profit distribution, and
14 automated assessments of production quality on a plant-by-plant basis.

15 The major advantages supporting DDNS design include:

- 16 1. Machine interactions are name lookups, not actual data transfers, until the very
17 last moment when a site-to-site connection can be made. Thus, interactions
18 are very fast between machines.
- 19 2. Information contained on any machine emphasizes minimalist storage. A
20 machine attempts to keep only the most minimal information it needs on its
21 local system, knowing that it can retrieve remote information very quickly
22 whenever needed.
- 23 3. Storage of data is genuine distributed. Since DDNS servers only hold index
24 information for where to get information, rather than the information itself,

global databases can be unlimited in size. Even very large global databases that exceed the physical storage possibilities of single sites will have excellent performance.

To illustrate the benefits, consider an example in which local sites cannot usefully store more than 50 terabytes of information. Since a global database supported by SDTP/DDNS only stores addressing information, it can provide information on many such sites, letting those sites resolve the actual data internally.

4. Each machine may cache its previous lookups and thus avoiding vertical lookups for recurrently used data. For data that sister sites will share over and over, the caching mechanism allows site-to-site connections without making parent-node queries. Thus, performance is very fast, even when scaling to very large addressing mechanisms.

5. Local policies determine disk storage and internal data structure for local systems. Yet local information will be available globally.

Flexibility & Ease of Use

Flexibility, generality, and ubiquitous accessibility are core principles of the SDTP/DDNS implementation. With a minimal "backbone" infrastructure of a small collection of machines, DDNS can support numerous concurrent universal databases, conveniently supporting as diverse systems as automated parts tracking in automobile repairs, insurance records, and purchase and repurchase of any scanable item: clothing, home appliances, wood, paint, groceries, etc. Such applications will only require a barcode click on behalf of the user. And then on behalf of that user, a computer queries a DDNS server for data location, and then SDTP retrieves the data from the appropriate site.

Simple Data Transport Protocol (SDTP)

1 The following terminology and associated definitions are used in this specification:

2 **Database:** A collection of records. SDTP supported databases can be local and/or
3 global.

4 **Record:** A denotatum stored in a database.

5 **Transaction:** An interaction between a client query and server response. Example
6 transactions include modifying a global database and finding a record
7 in a database.

8 **Client Query:** A request from a client sent to a server.

9 **Server Response:** A response from a server sent to a client.

10 **Message:** Generally, a client query or server response. Specifically, a content
11 identifier and data object.

12 **Content Identifier:** The first string of a SDTP message that identifies SDTP/0.9
13 version, command, and arguments to process.

14 **Data Object:** A header and body. Data objects may include text, images, video,
15 sound, etc.

16 **Header:** Header information in a data object, employing MIME conventions.

17 **Body:** Body information in a data object, employing MIME conventions.

18 The Simple Data Transport Protocol (SDTP) of the present invention is a protocol that
19 applies to dynamically distributed, Internet-wide, database systems.

20 The SDTP protocol provides universal addressing of database systems, and universal
21 search and retrieval of data stored on such systems. SDTP supports any encoding
22 mechanism, but is optimized for large scale or universal encoding mechanisms for universal
23 image tracking.

24 SDTP distributed database functionality can seamlessly traverse any network

1 topology, machine-type, operating system, database system. etc. The protocol supports all
2 forms of data: text, image, video, sound, etc.

3 SDTP permits intelligent, efficient, fully automated data-sharing on a site-to-site,
4 device-by-device basis, searching and retrieving specific data sets. Data sets can be located
5 anywhere on a network, and have no physical storage-size restrictions. Searching can be
6 local or global. For example, data produced by an ECG machine in Chicago is available to
7 another ECG machine in Bangkok.

8 In SDTP data are no longer viewed as existing on a single system, or any collection of
9 explicitly linked systems or sub-systems, such as credit card authorization systems. Rather,
10 SDTP views data as query relations, in which a single, very simple query mechanism
11 dynamically organizes and retrieves germane information at a moment of request.

12 The basic mechanism works such that when a client query is fully satisfied locally, a
13 search can halt. When a client query is not satisfied locally, within a couple of network
14 "hops" a SDTP server response will return a comprehensive list of data locations to the
15 requesting client. A given client needs no initial knowledge that related data even exist, or
16 where or how such data are stored. Yet for any given record, a client query can rapidly find
17 all related records across the Internet-even if related records exist on databases unknown to
18 exist by the requesting client, at the moment of its request.

19 SDTP can support machine clusters, LANs, WANs, heterogeneous networks,
20 collections of linked networks, or any set of these. This design explicitly includes support for
21 full, Internet-wide search and retrieval of records. Essentially, there are no network
22 restrictions for SDTP; it can transport and retrieve information for local or global systems
23 alike.

24 SDTP operations rely on client-server transactions. A transaction is characterized by

a client sending a message to a server, and the server sending back a message to the client.

SDTP/0.9 supports two basic transactions, Lookup and Modify, each of which has two

commands. Table 2 summarizes these relations.

Lookup	index query	Retrieve a record's index, but not the record itself
		Retrieve a record.
ModifyADD		Add a record to a database.
	DELETE	Remove a record from a database.

Table 2: Transactions

Since SDTP applies to any kind of database, existing anywhere on the Internet, SDTP transactions provide genuinely global searching, retrieving, adding, and removing records from universal databases.

As noted earlier, a message is a client query or a server response. The data structure of messages consists in a content identifier and data object. Table 3 summarizes these relations. A content identifier identifies SDTP version, command, and any arguments. A data object

Content Identifier	
Data Object	Header
	Body

Table 3: Message Data Structure

consists in a header and body, both of which support MIME conventions. A header contains information about the transaction-type and data specifications. A body contains data, which can include MIME multipart documents, among other data.

SDTP relies on uniform interaction between clients and servers, transacted through client query and server response messages. A client query requests actions from a server. A server response answers client queries, and sometimes also performs actions on behalf of the

client called server actions. Both client queries and server responses rely on the data structure of messages.

Table 4 summarizes protocol relations. Since transactions are interactions between client queries and server responses, a 'Lookup index' transaction would involve an exchange between a client query and a server response. In turn, client queries and server responses are subject to content identifiers and data objects.

Transactions	Lookup	index
		query
	Modify	ADD
		DELETE
Client Query	Content Identifier	
	Data Object	Header
		Body
Server Response	Content Identifier	
	Data Object	Header
		Body

Table 4: Protocol Relations

Message Data Structure

A message is a content identifier and a data object. The SDTP client query content identifier syntax is:

SDTP/version command argument ...

These expansions describe content identifier semantics:

version	SDTP version number.
command	Keyword specifying SDTP activity.
argument	Argument for command.
...	Subsequent arguments.

Example. Consider this legal content identifier:

SDTP/0.9 LOOKUP MedImages 123.abc

Following the content identifier syntax, the command is 'LOOKUP' with two arguments: (1)'MedImages' (the database to search) and (2) '123.abc' (a record or set of

records as might occur in a hospital system).

A data object includes (1) a header and (2) a body. A data object may include text, images, video, sound, any other media or data type, and any combination thereof.

A header consists in one or more lines and is terminated by a blank line. The syntax for such lines is:

fieldname: data

The argument data may have any number of additional arguments separated by semicolons (;). These expansions describe the header semantics:

fieldname	Label for data field	(e.g., Date, Content-Length, Content-Type, etc.)
data	Data	(e.g., for content-type, cookies etc.)

Client query data object headers additionally can contain preferences for server responses, but SDTP/0.9 does not yet specify these.

For example, consider the following data object header. This example illustrates a Lookup transaction that retrieves a record:

Content-Type: application/sdtp; transaction="lookup-1"; lookuptype="query"

A body consists in a MIME body, including multipart bodies [FB96a]. This structure facilitates transmission of all data types such as text, graphics, sound, video, etc.

A body contains content or is null. The exact format of the body depends upon specific databases, and thus is fixed in a separate standardization process not subject to SDTP.

For example, consider the following data object body. This example indicates a successful deletion of record '123.abc':

SUCCEEDED DELETE 123.abc

A client query is a message, and consists in (1) a content identifier and (2) a data

1 object.

2 An example of a client query data object is:

3 Content-Type: application/sdtp; transaction="modification"

4 From: medical.cenon.com

5
6 DELETE 123.abc

7 ADD DEF0456

8
9 A server response is a message, and consists in (1) always a data object, and (2)
10 sometimes an additional server action (e.g., a record deletion). See also Message Data
11 Structure.

12 Unlike client queries, SDTP server responses have no content identifier. Server
13 responses are data objects.

14 Server response data object headers are transaction types.

15 For example, consider the following server response data object header. The
16 Transaction type illustrates address forwarding.

17 Content-Type: application/sdtp; transaction="forwarding"

18 The Server response data object body syntax is determined in the data object header
19 according to the transaction specification. A body contains content or is null. For example,
20 consider the following server response data object body which illustrates successful deletion
21 of record '123. abc', and failed addition of record 'DEF@456'.

22 SUCCEEDED DELETE 123.abc

23 FAILED ADD DF.FO456

24
25 Example Server Response Data Object

26 Consider this example of a server response data object, including both header and
27 body:

28 Content-Type: application/sdtp; transaction="modification"

29 From: medical.cenon.com

SUCCEEDED DELETE 123.abc
 FAILED ADD DEF@456

A server response may invoke a server action, such as adding or deleting a record from a database. SDTP specifies the structure and syntax of data objects. SDTP specifies a semantics associated with the server action, but not structure or detail of implementation. Such considerations are left to decisions of site-by-site implementation.

Transactions

A transaction consists of (1) a client query (to server), and (2) a server response (to client). The current version, SDTP 0.9, provides two types of transactions, Lookup and Modify. Table 5 illustrates these relations.

	Client Query	Server Response
Lookup	Client queries server for records.	Server returns records, or location instructions for where to find records.
Modify	Client requests server synchronization	Server synchronizes data storage.

Table 5: Transaction Summary

Lookup: A Lookup transaction determines if a node has knowledge of requested record(s).

Client Query: A Lookup client query is a message, and thus contains (1) a content identifier and (2) a data object.

(1) Content Identifier: An sample Lookup content identifier is as follows:

SDTP/0.9 MODIFY MedImages 123.abc

(2) Data Object: A Modify data object will contain (A) a header and (B) a body.

(2A) Header: The Lookup header uses a Content-Type that specifies (1) a Lookup transaction and (2) a lookuptype. A Lookup data object header has syntax:

Content-type: application/sdtp; transaction="lookup"; lookuptype="type" "type" in

'lookuptype' has these values and meanings in SDTP/0.9:

Value	Meaning
query	Transfer a record
index	Transfer a record's index, but not the record itself

The following example of a Content-Type data object header retrieves a record's index, but not the record itself:

Content-Type: application/sdtp; transaction="lookup"; lookuptype="-index"

(2B) Body: A Lookup client query data object body is empty in SDTP/0.9.

Server Response: A Lookup response (1) has no content identifier, and (2) has a data object.

(1) Content Identifier: The Lookup server response has no content identifier.

(2) Data Object: The Lookup server response data object has (A) a header and (B) a body.

(2A) Header: This header specifies the Content-Type of the server response, but may also include other information such as MD5 encrypted signatures, etc.

(2B) Body: The body follows MIME message body conventions [FB96a]. SDTP data object bodies have three forms:

1. One or more records, structured as MIME multipart documents [FB96a].
2. Forwarding instructions, for one or more records, structured as the MIME type application/sdtp with attribute transaction set to forwarding. For example:

Content-Type: application/sdtp; transaction="forwarding"

A following data object body would contain a list of addresses to query for records.

3. Compound responses, structured as MIME multipart documents. Each part of a multipart document will be:

- (a) Form 1, One or more records
- (b) Form 2, Forwarding instructions

(c) Form 3, Compound responses.

Example Lookup Transaction: This example illustrates a Lookup transaction, in which record '123.abc' is retrieved from universal database 'MedImages'. The transaction consists in a client query and a server response.

Client Query: The following 3 line transcript is a plausible client query Lookup record retrieval request, from medical.cenon.com.

```
SDTP/0.9 LOOKUP MedImages 123.abc
Content-Type: application/sdtp; transaction="lookup"; lookuptype="query"
From: medical.cenon.com
```

This query requests the retrieval of record '123.abc' from universal database 'MedImages'.

Server Response: The following 7 line server response indicates a plausible server reply to the previous client query.

```
SDTP/0.9 LOOKUP MedImages
Content-Type: application/sdtp; transaction="forwarding"
From: medical.cenon.com

ddns-2.uch.net
ddns-2.mag.net
ddns-2.nyc.net
```

The server forwards the client addresses where questions are answered about record '123.abc' in universal database 'MedImages'.

Modify: A Modify transaction synchronizes databases. A client query asks for modification of server-stored data, such as adding or deleting a record.

Client Query: A Modify client query includes (1) a content identifier and (2) a data object.

(1) Content Identifier

See Content Identifier about content identifier syntax and semantics. An example Modify content identifier looks like:

SDTP/0.9 MODIFY MedImages 123.abc

(2) Data Object

See Data Object about data object syntax and semantics. A Modify data object will contain (A) a header and (B) a body.

(2A) Header. A Modify data object header specifies a Content-Type using transaction="modification". An example looks like:

Content-Type: application/sdtp; transaction="modification"

(2B) Body. The Modify data object body contains instructions detailing modification of the database, such as adding and deleting records. An example data object body request for deleting record '123.abc' and adding record 'DEF@456' could be:

DELETE 123.abc
ADD DEF@456

Server Response

A Modify server response (1) has no content identifier, and (2) has a data object.

(1) Content Identifier The Modify server response has no content identifier.

(2) Data Object

See Data Object about data object syntax and semantics. The Lookup server response data object has (A) a header and (B) a body.

(2A) Header. This header specifies the Content-Type of the server response, but may also include other information such as MD5 encrypted signatures, etc.

(2B) Body. The body follows MIME message body conventions [FB96a]. The Modify data object body may also contain verification information, such as the success or failure of a request.

Verification information often is null.

Example Modify Transaction

1 The following example illustrates a Modify transaction. This example modifies the
2 universal MedImages database, where the client query requests to delete record 123.abc and to
3 add record DEF@456. The transaction consists in a client query and a server response.

4 Client Query. The following 6 line transcript requests to delete and add a record from
5 database MedImages.

```
6           SDTP/0.9 MODIFY MedImages
7           Content-Type: application/sdtp; transaction="modification"
8           From: medical.cenon.com
9
10          DELETE 123.abc
11          ADD DEF@456
```

12
13 Line 1. Client query identifies protocol and requests modification of database MedImages.

14 Line 2. 'Content-Type' identifies a SDTP application; 'transaction' specifies a "modification"
15 transaction type.

16 Line 3. 'From' identifies client making request.

17 Line 4. Blank line identifies separation between data object header and data object body.

18 Line 5. 'DELETE' removes forwarding for Lookup query of record '123.abc'.

19 Line 6. 'ADD' enables forwarding to 'medical.cenon.com', for Lookup query of record
20 'DEF@456'.

21 Server Response. The following 6 line server response indicates a plausible reply to the previous
22 client query.

```
23           SDTP/0.9 MODIFY MedImages
24           Content-Type: application/sdtp; transaction="modification"
25           From: medical.cenon.com
26
27          SUCCEEDED DELETE 123.abc
28          FAILED ADD DEF@456
```

29
30 Line 1. Server response identifies protocol and acknowledges request for modification
31 of database MedImages.

Line 2. 'Content-Type' identifies a SDTP application; 'trisection' specifies a "modification" transaction type.

Line 3. 'From' identifies client making request.

Line 4. Blank line identifies separation between data object header and data object body.

Line 5. 'SUCCEEDED' indicates that client query 'DELETE' was completed successfully.

Line 6. 'FAILED' indicates that client query 'ADD' was not completed successfully.

Referring to figure 1 a simplified DDNS Nodal indexing system is illustrated. If the electrocardiogram analyzer 10 has unique identifier which, for purposes of this illustration is noted as the number one. When a user first is established with the system of the present invention the user might be subject to electrocardiogram analyzer testing on, for example, January 11th 1998. This date, in combination with unique electrocardiogram analyzer identifier "one" is then registered was in DDNS server 14 and. In this example the DDNS server is noted as a Level-2 server located at the University of Chicago noted with the abbreviation "UCH."

Other electrocardiogram analyzers 12, 16, and 18 are also shown as part of the system. Electrocardiogram analyzer 12 is connected to DDNS server 14. Electrocardiogram analyzer 16 is connected to DDNS server 20 which, in the present example is noted as being associated with Massachusetts general hospital, abbreviated as "MAG.". Electrocardiogram analyzer 18 having the unique identifier "4" is connected to DDNS server 22 in New York City.

DDNS level 2 servers 22, 20, and 14 are connected to a DDNS Level-1 server 24. The purpose of the DDNS Level-1 server 24 is to receive and store a record of the identifiers of individual date or records created by the individual electrocardiogram analyzers shown in figure 1. It is important to note that the DDNS Level-1 server 24 does not contain the ultimate information, that is, the results of electrocardiograms that have been administered to the particular patient at the various hospitals. (It should be noted that DDNS level 1 server may

1 actually be several machines as is later illustrated.) DDNS Level-1 server 24 only contains a
2 record of the indices that show where the information is stored. Thus for example, if a patient
3 receives an electrocardiogram on January 11th 1998 from electrocardiogram analyzer can the
4 information record concerning that analysis is stored and the user is given a medical
5 identification card possessing and ID number associated with at first examination. If that patient
6 is subsequently examined using electrocardiogram analyzer 18 on September 23rd 1998, the
7 practitioner would obtain the medical identification card of the patient, record that information
8 through electrocardiogram analyzer 18 which would then determine if it had ever seen that
9 particular patient before. If not, a query would proceed from electrocardiogram analyzer 18 to
10 DDNS Level-2 server 22 to inquire if a record of that patient exists. If no such record exists at
11 DDNS Level-2 server 22 that server will send a query to DDNS Level-1 server 24 to determine
12 if it has a record of the particular patient.

13 DDNS Level-1 server or 24 will have such a record of the existence of the particular
14 patient is existing at electrocardiogram analyzer 10 which can be reached via DDNS server 14.
15 Thereafter DDNS Level-2 22 will make contact with DDNS Level-2 server 14 on behalf of
16 electrocardiogram analyzer 18, with electrocardiogram analyzer 10.

17 In this example, DDNS servers of two different levels are shown. The level 2 servers
18 store and broker requests for indices relating to medical equipment that is connected to them.
19 The level 1 DDNS servers store and broker requests for indices relating to patients from Level
20 2 servers connected to them. Thus information is not generally passed when a query is made,
21 only the identification of the location of the data is transferred. It should also be noted that this
22 example of use in the medical arena is not meant to be limiting. As will be explained later, other
23 application areas are equally considered to be within the scope of the present invention.

24 Referring to figure 2 the network topology of the present invention is illustrated. As

1 noted earlier, the present invention is device driven rather than driven by a top-down
2 classification schemata. A series of medical devices may be attached to a workstation at a
3 particular hospital. For sample, ECG 4, CT 6, and ECG 12 may all be attached to a workstation
4 2 at a location, for example, the University of Chicago (UCH). ECG 4 will have its own unique
5 identifier generally assigned by the manufacturer. When the ECG 4 is first activated to perform
6 its first examination and creates the appropriate patient record, ECH 4 becomes registered on the
7 system of the present invention. Thus the global identifier for ECG 4 is created the first time the
8 device is activated. In a similar fashion, CT 6 also has a unique identifier as does ECG 12.
9 These devices are also shown as attached to workstation 2.

10 Alternatively medical devices may be self-contained and amenable to being attached or
11 directly connected to a DDNS server. The situation is also illustrated in figure 2 where CT 8 is
12 shown as directly connected to second level DDNS 14. Additionally, ECG 10 is also shown as
13 connected to the DDNS level 2 server 14.

14 All of the above devices ECG 4, CT 6, ECG 12, CT 8, ECG 10, and workstation 2 are
15 all uniquely identified and registered with the system of the present invention the moment that
16 they are first activated. As will be shown later, a date time stamp is also used in conjunction with
17 the unique identifier to create a unique patient identifier the first time that the patient uses any
18 device that is registered on the system.

19 In a similar fashion, other medical devices and other geographically disparate locations
20 can also become registered on the system of the present invention. Again referring to Figure 2
21 ECG devices 13, 15, and 16 may all be resident at, for example, Massachusetts General Hospital
22 (in this figure designated as MAG). Additionally, MRI 11 and CT are also shown as located in
23 Massachusetts General hospital. ECG 15 and 16 may be located in emergency room 21 while
24 ECG 13 may be located in and the attached to a workstation and cardiology 19. MRI 11 and CT

1 9 may be connected to workstation 17 in radiology. Workstations 21 in the emergency room, 19
2 in cardiology, and 17 in radiology are all connected to DDNS level 2 server 20. It is important
3 to note that unique identifiers will only be associated with the devices actually performing the
4 diagnostic task, that is, devices 9,11, 13,15, and 16. Workstations 17, 19, and 21 will not to have
5 unique identifiers since they will not be creating the medical records to be searched. It is of
6 course possible that an individual hospital may wish to have unique identifiers regarding all
7 devices on a network whether diagnostic or not in order to be able trace all instances of data or
8 information created. The present invention will support this implementation as well.

9 In Figure 2 yet a third location, hypothetically a hospital in New York City
10 (designated as NYC) is shown as having a series of medical devices as well. In this is an
11 instance ECG 18 is connected to a PC 32. That PC is in turn connected to a display 30 was
12 capable of displaying the results of the ECG analysis. Similarly, ECG 34 is directly
13 connected to display 30. Display 30 is connected to a workstation 26 which has an MRI
14 device 28 connected to it. This workstation 26 is in turn attached to DDNS level-2 server 22.

15 All of the DDNS level-2 servers 14,20, and 22 are connected to DDNS level-1 servers
16 23, 24, and 25. These DDNS level-1 servers broker queries for information and client index
17 locations coming from the various geographic locations were a patient may be treated.

18 It is important to note that once a medical device is activated for the first time, its unique
19 ID is stored at both the DDNS level-2 server and a DDNS level-1 server. This is because the
20 DDNS Level-2 server knows about diagnostic devices connected to it and DDNS Level-1 servers
21 know about DDNS Level-2 connected to them. Thus the DDNS servers learn about the
22 diagnostic devices in different ways. Further, once a patient is treated for the first time using
23 any medical device that is attached to the present system, a permanent designation for that patient
24 is created which comprises the unique identifier of the medical device combined with the date

1 time stamp of the first treatment of the patient.

2 In practice, and as will be discussed in detail later, a patient who is treated at ECG 18 in
3 NYC, and who possesses a medical ID card or barcode label on any card created by the system
4 will cause a query to be created to determine if any other medical records exist for the patient.
5 Initially a query will go as high as DDNS level-2 server 22. If an index to that client's records
6 exists within the New York City location, that information will be sent to the operator of ECG
7 18. If such information does not exist, DDNS level-2 server 22 will add a new record to its
8 database and will send the query to DDNS level-1 servers 23, 24, and 25 to determine where a
9 patient index for that particular patient does exist. If DDNS Level -1 server knows about the
10 record, it will make a record associating the the new data record with the data record that already
11 exists in its database. If the patient was first treated at University of Chicago, the patient index,
12 derived from the medical ID card, will cause a query to be sent to DDNS level-2 server 14 which
13 will respond with the various indices which indicate that location of records relating to the
14 patient of interest.

15 Examples

16 By way of example and to further illustrate a preferred embodiment of the present
17 invention, Figure 3 is presented. Since only data regarding indices to information os being
18 searched, the system of the present invention responds very quickly to queries for the location
19 of patient information. In the examples that follow, only the date@device designation is used.
20 In the full implementation the manufacturers ID would also be present as part of the unique
21 identifier.

22 1. Jane comes the University of Chicago Hospital on 11 December 1998 for the first time,
23 to receive her first ECG ever, where she receives a University of Chicago Hospital medical card
24 as a normal part of her admission. Upon receiving an ECG exam, Jane's reading is automatically

1 entered into the University of Chicago Hospital's local database system, and a printer produces
2 a small sticker which a nurse affixed to Jane's medical card.

3 SDTP/0.9 MODIFY MedImages
4 Content-Type; application/sdtp; transaction="modification"
5 From; ddns-2.uch.net

6
7 ADD 19981211@1
8

9 Since DDDS-2:UCH has never seen record '19981211@1', it attempts to 'ADD' it to the
10 next level "up" (to DDNS-1 Servers). A similar SDTP client query is sent "up" to DDNS-1
11 Servers.

12 2. DDNS-1 Servers receive and process the following client query request to ADD record
13 '19981211@1' to the global 'MedImages' database:

14 SDTP/0.9 MODIFY MedImages
15 Content-Type; application/sdtp; transaction='modification'
16 From: ddns-2.uch.edu

17
18 ADD 19981211@1
19

20 Since DDNS-1 Servers have not yet seen record '19981211@1' DDNS-1 Servers store
21 it. Using the From: field, DDNS-1 Servers further associate '19981211@1' with address 'ddns-
22 2.uch.edu.' DDNS-1 Servers will now forward future Lookup requests for record '19981211@1'
23 to DDNS-2:UCH.

24 DDNS-1 Servers return a SDTP server response to DDNS-2:UCH. The response
25 indicates successful completion of the request ADD for '19981211@1' in database 'MedImages.'

26 The server response is:

27 SDTP/0.9 MODIFY MedImages
28 Content-Type: application/sdtp; transaction="modification"
29 From: ddns-2 uch.edu
30 SUCCEEDED ADD 19981211@1
31

32 The DDNS system has now globally registered record '19981211@1' in the universal database

1 'MedImages'.

2 Crucial implications follow from this design

3 (a) A local system creates a global identifier in a fully automatic manner, without any human
4 intervention whatsoever.

5 (b) This process requires no form of "root registration" for users of equipment.

6 (c) A user may operate whatever local system is most desirable, without interference from
7 SDTP/DDNS.

8 3. Jane returns for a follow-up visit on 22 Dec 1998. A nurse clicks a barcode reader on
9 Jane's medical card, which concurrently prompts DDNS-2:UCH to do an internal 'Lookup' client
10 query for the number encoded onto her card ('19981211@1') which was encoded during her first
11 visit on 11 Dec 1998. Since DDNS-2:UCH has the record '19981211@1' it stops searching, and
12 does not query DDNS-1 Servers. Jane's previous reading of 11 Dec 1998 is automatically
13 recalled onto the screen.

14 4. The 'ADD' command now saves the 22 Dec 1998 reading, associating it with the 11 Dec 1998
15 reading through the global index (database "key") '19981211@1'. The SDTP interaction looks
16 like:

17 SDTP/0.9 MODIFY MedImages
18 Convent-Type: application/sdtp; transaction="modification"
19 From: ddns-2.uch.edu

20
21 ADD 19981211@1
22

23 The attending physician refers Jane to a Networked Specialist Physician, who makes an
24 appointment of 5 Jan 1999.

25 Crucial implications follow from the client-server interactions up to now:

26 (a) DDNS-1 Servers store a mapping from record '19981211@1' to DDNS-2:UCH. DDNS-
27 2UCH has a mapping to whatever internal mechanism the University of Chicago Hospital uses

1 to record its data.

2 (b) DDNS-1 Servers and DDNS-2:UCH only know about one record ('19981211@1'), while
3 the University of Chicago Hospital knows about two records. A "one-to-many" relationship is
4 created.

5 Because only 1 number is stored at the DDNS-1 Servers level, global system performance
6 is optimized. Only 1 number provides potential access to all of Jane's records at the University
7 of Chicago Hospital.

8 (c) Routine business at the University of Chicago Hospital remains on site. This reduces
9 system complexity on the local side, and further optimizes Level-1 DDNS performance. DDNS
10 uses global resources only when local resources do not have the needed information.

11 (d) A user participates in global information sharing without any special interaction besides
12 barcode reading a medical card. Currently, optical readers provide the most robust construction,
13 but any other encoding mechanism could be used. Similarly, rather than affixing a label, the
14 medical card be given at the end of the visit, with the universal identifier indelibly marked on the
15 card.

16 5. On 5 Jan 1999 Jane arrives for her referral appointment with a Networked Specialist
17 Physician (NSP) made during her last visit to the University of Chicago Hospital on 22 Dec
18 1998.

19 A nurse at the NSP barcode-reads Jane's University of Chicago Hospital medical card
20 to begin the process. The NSP local machine does an internal client query 'Lookup' for the
21 number encoded onto Jane's card ('19981211@1'). This record is not found on the local
22 machine, and so the machine sends a client query to DDNS-2:UCH.

23 SDTP/0.9 LOOKUP MedImages 19981211@1

24 6. DDNS-2:UCH receives the previous client query and does an internal 'Lookup' for

1 '19981211@1', which it finds. Since the search for record '19981211@1' is satisfied, DDNS-
2 2:UCH does not query DDNS-1 Servers. DDNS-2:UCH returns records from 11 Dec 1998 and
3 22 Dec 1998.

4 Although miles from the University of Chicago Hospital, clicking a barcode reader on
5 Jane's medical card provides the NSP instant retrieval of previous ECG readings at the
6 University of Chicago.

7 7. Although appearing to the SNP in the same moment, the system now ADDs the reading
8 taken on 5 Jan 1999 to the NSP local system, associating it with the global index '19981211@1'.

9 Since record '19981211@1' is stored for the first time on the NSP local machine, the NSP
10 local machine also attempts to ADD "up" record '19981211@1', for global registration in
11 database 'MedImages'. The NSP local system sends this command to its DDNS server, DDNS-
12 2:UCH:

13 SDTP/0.9 MODIFY MedImages
14 Convent-Type: application/sdtp; transaction="modification"
15 From: specialist.uch.edu
16
17 ADD 19981211@1
18

19 8. DDNS-2:UCH receives the previous client query 'ADD'. Since DDNS-2:UCH has
20 already registered record '19981211@1', it now associates the NSP address with record
21 '19981211@1', as an address answering questions about global record '19981211@1'. DDNS-
22 2:UCH returns a server response indicating the successful status of the client query ADD:

23 SDTP/0.9 MODIFY MedImages
24 Convent-Type: application/sdtp; transaction="modification"
25 From: specialist.uch.edu
26
27 ADD 19981211@1
28

29 9. On 6 Jun 1999 Jane visits Massachusetts General Hospital (MAG). A nurse barcode
30 reads Jane's University of Chicago medical card, which posts an internal Lookup client query to

1 DDNS-2:MAG, for the original ECG taken on 11 Dec 1998 (in Chicago, affixed to Jane's
2 University of Chicago Hospital card).

3 Not having seen record '19981211@1' before, DDNS-2:MAG asks if DDNS-1 Servers
4 know where to find information about this record. DDNS-2:MAG sends this client query
5 Lookup request to DDNS-1 Servers:

6 SDTP/0.9 LOOKUP MedImages 19981211@1

7 10. DDNS-1 Servers receive DDNS-2:MaG's previous client query Lookup for record
8 '19981211@1' in global database 'MedImages'. DDNS-1 Servers do know about record
9 '19981211@1': that DDNS-2:UCH answers questions about it.

10 11. DDNS-1 Servers send a server response indicating that DDNS-2:UCH answers queries
11 about record '19981211@1'. The forwarding message sent is:

12 SDTP/0.9 LOOKUP MedImages
13 Content-Type: application/sdtp; transaction="forwarding"
14
15 ddns-2.uch.edu
16

17 12. DDNS-2:MAG directly connects to DDNS-2:UCH, which queries machine
18 specialist.uch.edu, requesting records related to '19981211@1':

19 SDTP/0.9 LOOKUP MedImages 19981211@1

20 DDNS-2:UCH returns records for ECG readings taken on 5 Jan 1999, 22 Dec 1998, and
21 11 Dec 1998.

22 Barcode reading Jane's University of Chicago Hospital medical card in Massachusetts
23 General Hospital instantly retrieves Jane's previous records, displaying the images on the screen
24 within a moment of clicking the barcode reader.

25 13. Although appearing in the same moment from the nurse's point of view, the system now
26 saves the 6 Jun 1999 reading to the local Massachusetts General Hospital machine.

1 Since DDNS-2:MAG does not have the '11 Dec 1998' reading (globally indexed by
2 '19981211@1'), it adds '19981211@1' to its database, associating it with the new reading taken
3 locally on 6 Jun 1999. Since '19981211@1' is a new record to DDNS-2:MAG, it also attempts
4 to ADD "up" the record just scanned from Jane's University of Chicago Hospital medical card,
5 '19981211@1'. DDNS-2:MAG sends a client query ADD to DDNS-1 Servers.

6 SDTP/0.9 MODIFY MedImages
7 Content-Type: application/sdtp; transaction="modification"
8 From: ddns-2.mag.org

9
10 ADD 19981211@1

11
12 14. DDNS-1 Servers receive the previous client query request to ADD record '19981211@1'
13 to global database 'MedImages'. Since DDNS-1 Servers already know about record
14 '19981211@1', DDNS-1 Servers store the address in the From: header (ddns-2.mag.org) as a
15 location that will answer queries for global record '19981211@1'.

16 Future Lookup requests for record '19981211@1' now will receive forwarding
17 instructions for both DDNS-2:MAG and DDNS-2:UCH. DDNS-1 Servers now sends a server
18 response indicating status of the request:

19 SDTP/0.9 MODIFY MedImages
20 Content-Type: application/sdtp; transaction="modification"
21 From: ddns-2.mag.org

22
23 SUCCEEDED ADD 19981211@1

24
25 15. Jane is in a car accident in New York City, and is rushed to a random hospital, where an
26 attendant discovers a University of Chicago Hospital medical card in her purse.

27 Clicking a barcode reader on the card, a DDNS-2:NYC internal Lookup learns that record
28 '19981211@1' is unknown. So it sends a client query Lookup request "up", to DDNS-1 Servers.

29 16. DDNS-L Servers receive a client query Lookup request from DDNS-2:NYC:

30 SDTP/0.9 LOOKUP MedImages 19981211@1

1 DDNS-1 Servers know about record '19981211@1'. and have two forwarding address
2 associations: DDNS-2:MAG and DDNS-2:UCH.

3 17. DDNS-1 Servers know that DDNS-2:UCH and DDNS-2:MAG answer queries for record
4 '19981211@1', and return forwarding instructions for these addresses.

5 SDTP/0.9 LOOKUP MedImages
6 Content-Type: application/sdtp; transaction="forwarding"
7
8 ddns-2.mag.org
9 ddns-2.uch.edu
10

11 18. DDNS-2:NYC directly connects to DDNS-2:MAG and DDNS-2:UCH, querying for
12 record '19981211@1', using the following client query for both addresses:

13 SDTP/0.9 LOOKUP MedImages 19981211@1

14 Four records appear on the screen in the emergency room:

15 6 Jun 1999 DDNS-2:MAG
16 5 Jan 1999 DDNS-2:UCH
17 22 Dec 1998 DDNS-2:UCH
18 11 Dec 1998 DDNS-2:UCH
19

20 While the top-level DDNS service (DDNS-1 Servers) only knows about global record
21 '19981211@1', the New York Hospital emergency room now views four records, from three
22 devices, from two different sites, simply by barcode reading a University of Chicago Hospital
23 medical card.

24 The physicians need not know from which facilities Jane has records, and yet her
25 comprehensive ECG records are available instantaneously.

26 19. Although appearing to the emergency room in the same moment, the system now ADDs
27 the reading taken on 23 Sep 1999 to the local DDNS-2:NYC system, associating it with the
28 global index of '19981211@1'.

29 Since record '19981211@1' is stored for the first time on the DDNS-2:NYC, DDNS-

Of course, each address answers queries using its own network architecture, storage

2:NYC also attempts to ADD "up" record '19981211@1', for global registration in database 'MedImages'.

20. DDNS-1 Servers receives the following client query request to ADD record '19981211@1' to global database 'MedImages', from DDNS-2:NYC.

SDTP/0.9 MODIFY MedImages
Content-Type: application/sdtp; transaction="modification"
From: ddns-2.nyc.com

ADD 19981211@1

Since DDNS-1 Servers already know about record '19981211@1' (from DDNS-2:MAG and DDNS-2:UCH), DDNS-1 Servers add the address in the From: header (ddns-2. nyc.com) to the list of addresses that will answer queries for global record '19981211@1'. DDNS-1 Servers then sends a server response indicating status of DDNS-2:NYC's client query ADD request:

SDTP/0.9 MODIFY MedImages
Content-Type: application/sdtp; transaction="modification"
From: ddns-2.nyc.com

SUCCEEDED ADD 19981211@1

Having made this addition, DDNS-1 Servers will answer future Lookup requests for record '19981211@1' with forwarding instructions to:

DDNS-2:NYC
DDNS-2:MAG
DDNS-2:UCH.

21. While on vacation in the Florida Keys, away from convenient access to a hospital, Jane experiences chest pain. She goes to an Independent Practitioner (IP) on 8 Oct 1999, and presents her University of Chicago Hospital medical card, which the physician barcode reads.

The physician's local machine does an internal client query 'Lookup' for the number

3 a variety of IBM PC and compatibles are suitable to the task of the workstations attached to
4 the medical diagnostic equipment or as DDNS-Level-2 servers. All that is required is that
5 there be sufficient storage to store the indices noted. The DDNS level-1 servers can also be
6 IBM PC, Sun workstations or indeed any server capable of storing and retrieving information
7 and communicating that information via modems or otherwise to other servers or
8 workstations of the present invention. Thus the software giving rise to the unique identifiers
9 is stored at the lower levels of the system while software for storing, retrieving and associated
10 indices are typically located at the high levels servers of the system.

11 Further Examples:

- 12 • On a home computer, Jane clicks a barcode on the last grocery receipt, which
13 reproduces the same order as the week before. Arriving at the grocery store,
14 the order has been charged to her credit card, and is waiting for pick up.
- 15 • On a home computer John clicks a barcode reader on an old pair of athletic
16 shoes. A web page appears on which he is offered a choice to repurchase the
17 same shoes (same brand, size, color, etc.), from the same company. He
18 mouse-clicks on "Accept" and the same credit card is charged that originally
19 purchased the shoes. The shoes are delivered to his house the next day. One
20 barcode click and one mouse click consummate this interaction.
- 21 • A student goes to a book store needing several chapters of various books. He
22 selects a book of interest from the shelves, and using the bookstore's PC
23 "kiosk", clicks on Chapter 3. A web-page returns instantly, indicating that he
24 may purchase this chapter for \$1.83, for which he may use his credit card or

1 pay the cashier.

2 Unknown to the consumer, at the moment of clicking the barcode reader, the
3 publisher has been consulted and the publisher's royalties and bookstore's profit margins have
4 been combined in the \$1.83 purchase. Each month the publisher sends automatically
5 generated invoices to bookstores that itemize royalties due. Such a mechanism works just as
6 easily for images in publications, and journal articles, etc., and in libraries and photocopy
7 shops to enhance copyright protection.

- 8 • A car owner brings a car to a garage for a muffler repair. Clicking a barcode
9 reader, the mechanic learns that the muffler is in warranty. The mechanic
10 provides better service to the car owner, and receives automatic part
11 restocking from the manufacturer.

12 The manufacturer not only automatically tracks muffler repairs for a given model, but
13 also the precise plants from which the failed mufflers come. The manufacturer has an
14 automatic way to track part life-cycles, manufacturing defects, repair warranties, replacement
15 policies, etc. Yet such data gathering is accomplished in the process of normal automotive
16 repair.

17 On site, a furnace repairman clicks a barcode reader at a broken furnace part, a
18 cellular phone calls into a local service provider, which in turn recalls the repair history of the
19 furnace, and the part availability for repair. New parts can be ordered from the portable
20 decoder the repairman brings to on-site jobs.

21 The repair process is streamlined, permitting the repairman to order parts order
22 directly from the manufacturer, while on-site, through an on-site cellular connection to a local
23 service provider, by passing administrative overhead back at the home office.

24 Many other such examples apply, such automatic tracing of insurance records by an

1 insurance company; repurchases of home appliances, wood, paints. The entire process is
2 universally and automatically indexed at production or transaction time, supported by this
3 document's "device-based" approach to universal database construction.

4 A system and method for establishing and retrieving data based on global indices has
5 been described. As previously noted, although a medical application has been described, it
6 will be appreciated by those skilled in the art from reviewing the specification and the
7 examples given that many other embodiments are possible using the system with departing
8 from the scope of the invention as disclosed.

1 We claim:

2 1. A Method for Establishing and Retrieving Data Based on Global Indices comprising:
3 establishing a unique device ID for each of a plurality of data generating devices on a
4 network;

5 registering the unique device ID of each of the plurality of data generating devices on
6 the network on at least one server connected to the network when the data generating
7 equipment is first used on the network;

8 establishing a unique user ID for each user of the data generating devices when the
9 user uses one of the plurality of data generating devices for the first time; and

10 retrieving data generated by the plurality of data generating devices by searching for
11 instances of the unique user ID.

12 2. The Method for Establishing and Retrieving Data Based on Global Indices of claim 1
13 wherein establishing the unique user ID further comprises combining the device ID of
14 the data generating device being used by the user with a date/time stamp of the first
15 use by the user.

16 3. The Method for Establishing and Retrieving Data Based on Global Indices of claim 2
17 further comprising storing the unique user ID on a token given to the user.

18 4. The Method for Establishing and Retrieving Data Based on Global Indices of claim 3
19 further comprising the user using the token with the unique user ID for all subsequent
20 uses of any of the plurality of data generating devices.

21 5. The Method for Establishing and Retrieving Data Based on Global Indices of claim 1
22 wherein the data generated is medical data concerning the user.

23 6. The Method for Establishing and Retrieving Data Based on Global Indices of claim 1
24 wherein the data generated is commercial data.

1 7. A System for Establishing and Retrieving Data Based on Global Indices comprising:

2 a network;

3 a plurality of servers connected to the network for storing data and responding to
4 search requests;

5 a plurality of data generating devices connected to the servers, wherein each data
6 generating device has a unique ID that is registered with at least one server when the data
7 generating device is first used of the network;

8 unique user ID generator associated with each data generating device whereby a
9 unique user ID is established by combining the unique device ID with a date time stamp of
10 when the user first used any of the plurality of data generating devices on the network; and
11 search logic for searching for instances of the unique user ID on any of the plurality of
12 servers.

13 8. The System for Establishing and Retrieving Data Based on Global Indices of claim 7
14 wherein the data generating devices are medical data generating devices.

15 9. The System for Establishing and Retrieving Data Based on Global Indices of claim 7
16 further comprising tokens generated by each of the plurality of data generating devices
17 on which is stored each unique user ID.

18 10. The System for Establishing and Retrieving Data Based on Global Indices of claim 7
19 wherein the data generated is commercial data.

20 11. The System for Establishing and Retrieving Data Based on Global Indices of claim 9
21 wherein each of the plurality of data generating devices further comprises a token
22 reader for reading the unique user ID stored on the token of a user.

23 12. The System for Establishing and Retrieving Data Based on Global Indices of claim 7
24 further comprising data transport logic for transporting data generated from one data

1 generating device to another once the location of the data has been determined by the
2 search logic identifying instances of the unique user ID on any of the plurality of
3 servers.

4 13. A Method for Establishing and Retrieving Data Based on Global Indices comprising:
5 establishing a unique device ID for each of a plurality of data generating devices on a
6 network;

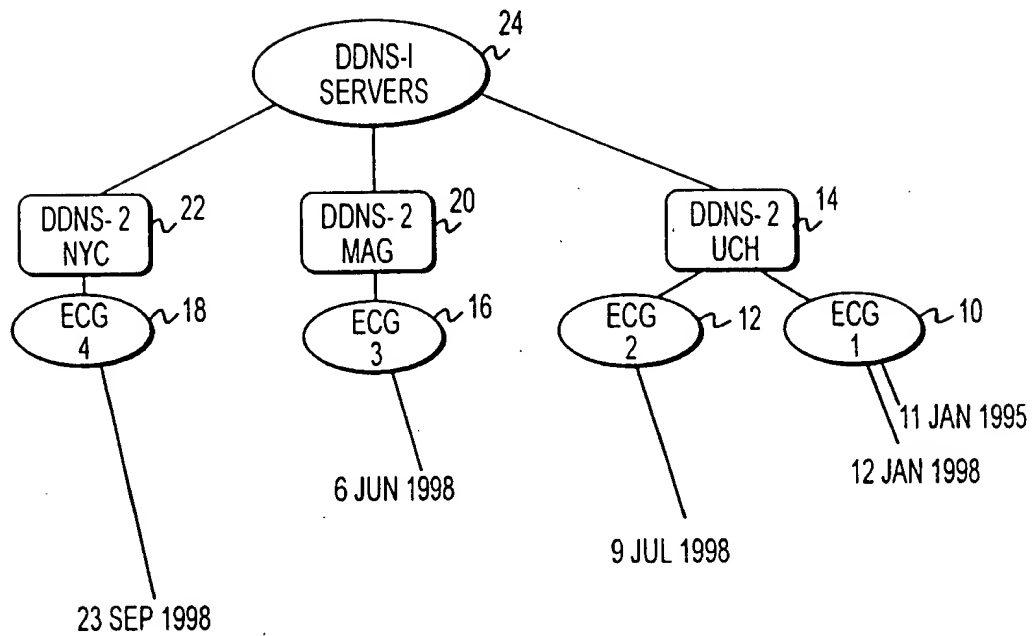
7 registering the unique device ID of each of the plurality of data generating devices on
8 the network on at least one server connected to the network when the data generating
9 equipment is first used on the network;

10 establishing a unique record ID for each record of the data generating devices when
11 the record is created using one of the plurality of data generating devices for the first time;
12 and

13 retrieving data generated by the plurality of data generating devices by searching for
14 instances of the unique record ID.

15 14. The method for establishing and retrieving data based on global indices of claim 13
16 wherein the records creating is creating records of parts of an assembly.

1/3

DDNS NODAL INDEXING
PRELIMINARY EXAMPLE
(FOR SINGLE AGENT)**FIG. 1**

2/3

SOTP/DDNS
NETWORK TOPOLOGY EXAMPLE

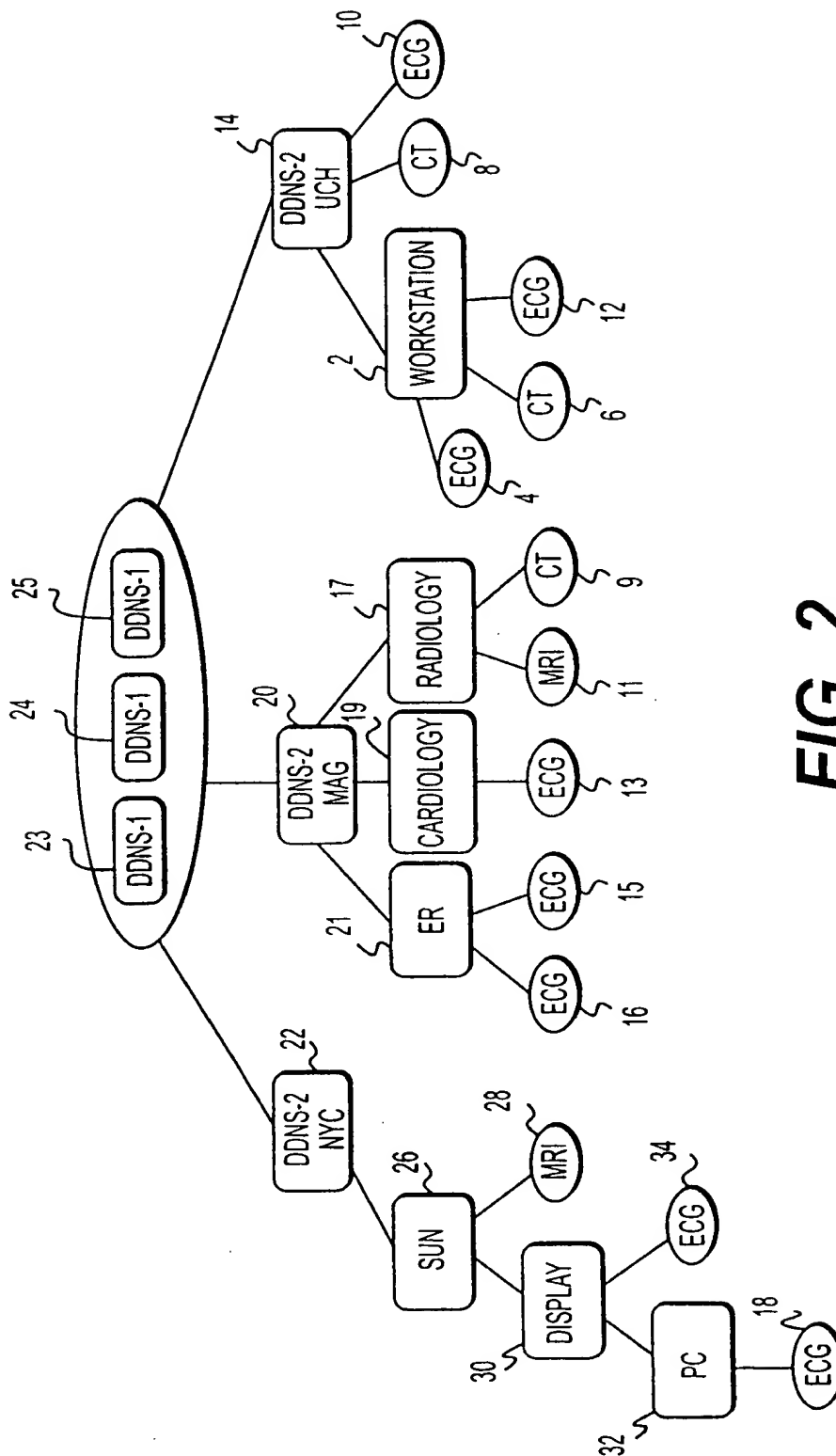


FIG. 2

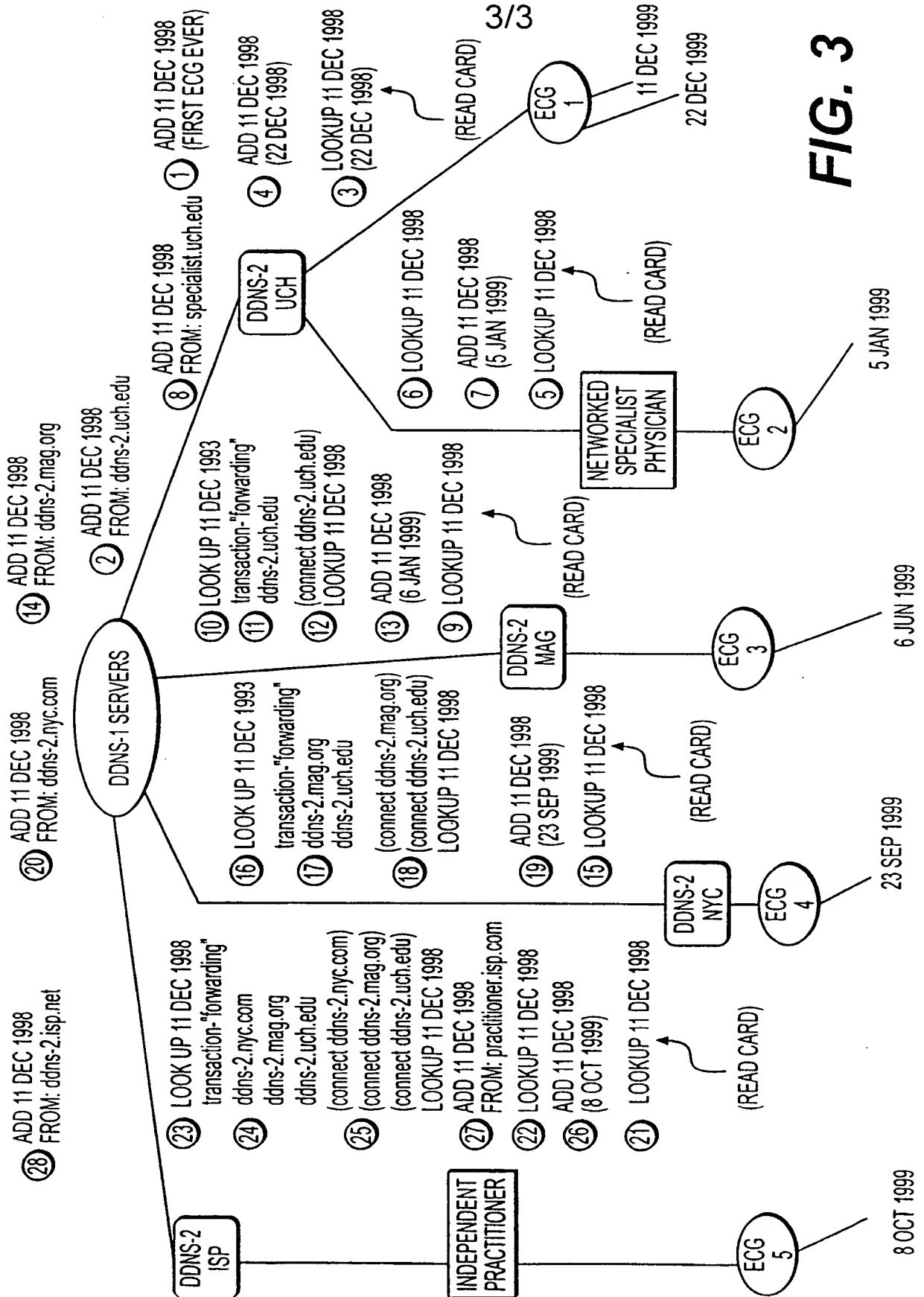


FIG. 3

INTERNATIONAL SEARCH REPORT

Inte. onal Application No
PCT/US 99/14553

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/12 G06F19/00 //G06F159:00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 98 15910 A (SCHULTZ JOSEPH PAUL ;SCHULTZ MYRON G (US)) 16 April 1998 (1998-04-16) page 3, line 12 -page 5, line 5 ---	1-5,7-9, 11-13
A	BOURKE D G ET AL: "PROGRAMMABLE MODULE AND CIRCUIT FOR MACHINE-READABLE UNIQUE SERIAL NUMBER" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 27, no. 4A, 1 September 1984 (1984-09-01), pages 1942-1944, XP000715172 ISSN: 0018-8689 the whole document --- -/--	1,7,13



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

19 October 1999

Date of mailing of the international search report

26/10/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Fournier, C

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/14553

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>KLEINHOLZ L ET AL: "SUPPORTING COOPERATIVE MEDICINE: THE BERMED PROJECT" IEEE MULTIMEDIA, vol. 1, no. 4, 21 December 1994 (1994-12-21), pages 44-53, XP000484150 ISSN: 1070-986X page 44, left-hand column, line 1 -page 47, left-hand column, line 11 -----</p>	1,5,7,8
A	<p>"METHOD FOR NETWORK NAMING AND ROUTING" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 37, no. 9, 1 September 1994 (1994-09-01), page 255/256 XP000473404 ISSN: 0018-8689 the whole document -----</p>	

Information on patent family members

PCT/US 99/14553

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9815910 A	16-04-1998	AU 4606597 A	05-05-1998

1-31-00

THIS PAGE BLANK (USPTO)